**Car Renting System**

TA: Alaa Khaled

Team: 22

# Introduction

The **Car Renting System** is designed to streamline the process of renting cars for customers and managing the fleet for administrators. The system allows customers to register, log in, search for available cars, rent cars, and cancel rentals. Administrators can manage the car inventory, including adding, updating, and removing cars, and be able to display available cars and rentals.

The system will integrate with an **Oracle Database** to store and manage data such as user information, car details, and rental records. It will be developed using **C# Windows Forms** to provide a user-friendly interface for both customers and administrators.

The system will streamline the car rental process, reduce manual errors, and enhance overall user experience. By leveraging modern technologies, it will be both user-friendly and easy to maintain, making it suitable for small to medium-sized car rental businesses.

# User Requirements

* Customers should be able to register and log in to the system.
* Customers should be able to view available cars and search for cars based on rental cost.
* Customers should be able to rent a car and cancel the rental before the rental period begins.
* Administrators should be able to manage the car inventory (add, update, or remove cars).
* Administrators should be able to display available cars and rentals.
* The system should validate user inputs and ensure data integrity.

# Functional Requirements

## User Registration

**Description:**

This function allows a new **customer** to create an account by providing personal details. The system validates the provided data and ensures compliance with the registration criteria before storing the new account in the database. (Admin has already stored account in database) **Inputs:**

* Email
* Password
* Confirm Password
* Phone Number
* Username
* Driver’s License Number
* Age **Source:**
* Provided by the customer through the registration form.

**Pre-conditions:**

* The customer must be at least **18 years old**.
* The driver’s license must be **valid and not expired**.
* The password and confirmation password must **match**.

**Post-conditions:**

* A new user account is created in the database.
* The customer is redirected to the main page.
* Generate new Id for the customer.

**Output:**

* A message box appears with the confirmation message:

*"Registration successful. You can now log in."*

## User Login / Logout

**Description:**

This function allows a registered customer or Admin to log in or log out of the system using their username and password.

**Inputs:**

* Username
* Password **Source:**
* Provided by the customer through the login form.

**Pre-conditions:**

* The customer must be **registered** in the system.
* The **entered username and password** must match the stored credentials in the database.

**Post-conditions:**

* Upon successful login, the main page is displayed.
* Upon logout, the session is terminated, and the login page is displayed.

**Output:**

* A message box appears:
* If successful: *"Login successful. Welcome!"*
* If failed: *"Invalid username or password. Please try again."*

## Display Available Cars

**Description:**

This function displays a list of all cars currently available for renting.

**Inputs:**

* Customer/Admin click the "View Available Cars" button.

**Source:**

* through the user interface.

**Pre-conditions:**

* There must be **at least one available car** in the database.

**Post-conditions:**

* The system retrieves and displays a list of available cars with relevant details.

**Output:**

* A list of available cars including:

Car ID, Model, Brand, Year, Rental Price Per Day, Availability Status.

## Search for Cars

**Description:**

This function allows customers to search for cars based on a specific rental cost.

**Inputs:**

* Rental cost entered by the customer.
* Customer clicks the "Search" button.

**Source:**

* Provided by the customer through the user interface.

**Pre-conditions:**

* There must be **cars available** in the database.
* At least one car must **match the specified rental cost**.

**Post-conditions:**

* The system filters and displays cars that match the entered cost.

**Output:**

* A list of cars that matches the specified rental cost, including:

Car ID, Model, Brand, Year, Rental Price Per Day.

## Rent a Car

**Description:**

This function allows a registered customer to rent a car by selecting a specific vehicle and rental duration.

**Inputs:**

* Car ID
* Customer ID
* Rental Start Date
* Rental End Date **Source:**
* Provided by the customer through the user interface.

**Pre-conditions:**

* The selected car must be **available for rental**.
* The customer must be **logged in**.

**Post-conditions:**

* The car’s status is updated to **"Rented"** in the database.

Generate new Id for rental.

* A new rental record is created.

**Output:**

* A message box appears:

*"Rental successfully created. Enjoy your ride!"*

## Cancel Rental (Discard Rent)

**Description:**

This function allows a customer to cancel an existing car rental before the rental period begins.

**Inputs:**

* Car ID
* Customer ID
* Rental ID **Source:**
* Provided by the customer through the user interface.

**Pre-conditions:**

* The **rental record must exist** in the database.
* The **rental period must not have started yet**.

**Post-conditions:**

* The car’s status is updated back to **"Available"**.
* The rental record is marked as **"Canceled"** in the database.

**Output:**

* A message box appears:

*"Rental successfully canceled."*

## Car Management (Admin Functionality)

**Description:**

This function allows the admin to manage the fleet of cars available for rental, including adding, updating, or removing cars.

**Inputs:**

* click on button "manage cars".

**Source:**

* Provided by the admin through admin interface.

**Pre-conditions:**

* The admin must be **logged** into the system.
* The entered **credentials must match** the stored admin credentials.

**Post-conditions:**

* The admin gains access to the **Car Management Dashboard**.

**Output:**

* Data Set contains all cars with their details.

## Display Rentals

**Description:**

This function displays a list of all rentals by Admin.

**Inputs:**

* Admin clicks the "View rentals" button.

**Source:**

* Triggered by the Admin from the user interface.

**Pre-conditions:**

The admin must be **logged in**.

**Post-conditions:**

* The system retrieves and displays a list of rentals with relevant details.

**Output:**

* A list of rentals information included:
* Rent id
* Car ID
* Customer id
* Cost
* Rent start time
* Rent end time

# Non-Functional Requirements

## Security (85%)

* The system will implement username and password authentication to prevent unauthorized access.
* Access control will be enforced:
* Only authorized personnel (admins) can access the admin dashboard.
* Only registered users with valid credentials can access the customer dashboard.

## Performance

• The system will ensure a response time of no more than **2 seconds** for user actions such as logging in, searching for cars, and renting a car.

## Availability

* The system will be available **24/7** for users to access and perform operations.
* In the event of a major system failure, the system will be restored within **1 to 2 working days** to minimize disruption to business operations.

## Error Handling (60%)

* The system will validate all user inputs to minimize errors.
* Appropriate error messages will be displayed to guide users in recovering from errors (e.g., "Invalid username or password" or "Please enter a valid rental date").

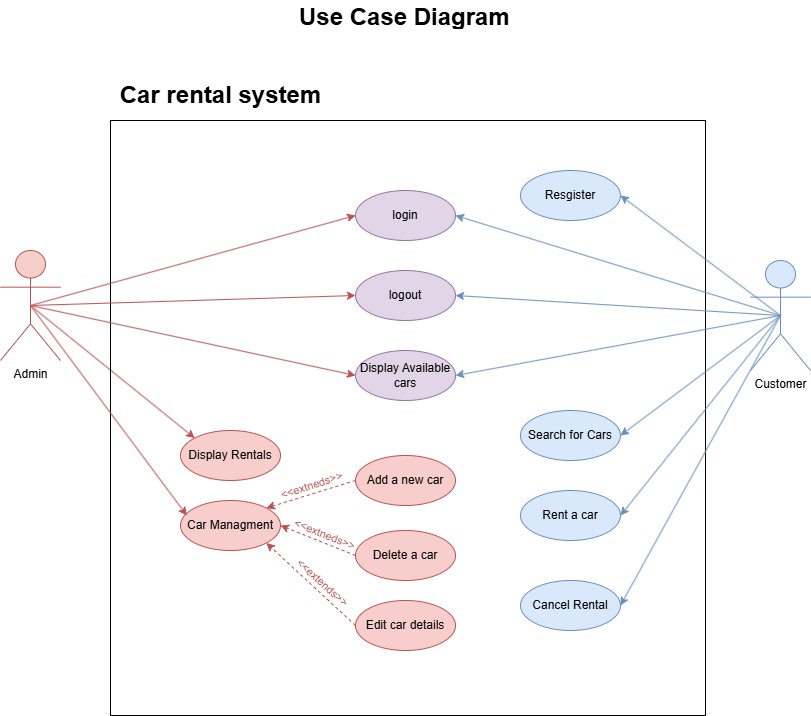
## Ease of Use (90%)

The user interface will be designed to be intuitive and user-friendly, requiring minimal training for end-users.

• Clear labels, instructions, and navigation elements will be provided to enhance usability.

## Development Environment and Tools

* **Programming Language:** C# windows forms
* **IDE:** Visual Studio 2022
* **Database:** Oracle Database
* **Version Control:** Git (with platforms like GitHub or GitLab for collaboration)



# Sequence Diagram of Car Rental Function

**Actors & Components:**

1. **Customer** – Initiates the car rental request.
2. **User Interface (UI)** – Collects input and sends it to the system.
3. **Rental Service** – Processes the rental request.
4. **Car Database** – Checks car availability and updates car status.
5. **Rental Database** – Creates a new rental record.

**Sequence of Events:**

**Customer -> UI: Enter Sign-Up Details**

**UI -> User Database: Check if Email Exists**

**User Database --> UI: Response (Exists/Not Exists)**

**UI -> User Database: Store New User Data**

**User Database --> UI: Confirm Account Creation**

**UI --> Customer: Display Success Message**

**Customer -> UI: Log in with Credentials**

**UI -> User Database: Authenticate User**

**User Database --> UI: Login Successful/Failed**

**Customer -> UI: Select Car & Enter Rental Details**

**UI -> Rental Service: Send Rental Request**

**Rental Service -> Car Database: Check Availability**

**Car Database --> Rental Service: Response (Available/Unavailable) Rental Service --> UI: Show Error if Unavailable**

**Rental Service -> Rental Database: Generate New Rental ID**

**Rental Service -> Car Database: Update Car Status to "Rented"**

**Rental Service -> Rental Database: Create Rental Record**

**Rental Service--> UI: Display Success Message**

